```
call void @llvm.dbg.value(metadata ptr %func args, metadata !78, metadata
                                                                    ...!DIExpression()), !dbg !86
                                                                     %call = tail call i32 @initialise arrays(ptr noundef nonnull @ func .s253)
                                                                     ... #8, !dbq !87
                                                                     %call1 = tail call i32 @gettimeofday(ptr noundef %func args, ptr noundef
                                                                    ... null) #8, !dbg !88
                                                                    call void @llvm.dbg.value(metadata i32 0, metadata !80, metadata
                                                                    ... !DIExpression()), !dbg !89
                                                                     br label %for.cond2.preheader, !dbg !90
                                                                      for.cond2.preheader:
                                                                      %nl.038 = phi i32 [ 0, %entry ], [ %inc21, %for.cond.cleanup4 ]
                                                                      call void @llvm.dbg.value(metadata i32 %nl.038, metadata !80, metadata
                                                                      ...!DIExpression()), !dbg !89
                                                                      call void @llvm.dbg.value(metadata i32 0, metadata !82, metadata
                                                                      ...!DIExpression()), !dbg !91
                                                                      %0 = call i32 @llvm.vscale.i32(), !dbg !92
                                                                      %extended.vscale = zext i32 %0 to i64
                                                                      %1 = shl i64 %extended.vscale, 2, !dbg !92
                                                                      %2 = icmp uge i64 32000, %1, !dbg !92
                                                                      br i1 %2, label %Pre.Vectorization, label
                                                                      ... %Preheader.for.remaining.iterations, !dbg !92
                                                                                       Τ
                                                                                                                             F
                                   Pre. Vectorization:
                                    %7 = \text{call i32} \otimes \text{llvm.vscale.i32}()
                                    %extended.vscale1 = zext i32 %7 to i64
                                    \%8 = \text{call} < \text{vscale x 4 x i64} > \text{@llvm.experimental.stepvector.nxv4i64()}
                                    %step.value = shl i64 %extended.vscale1, 2
                                    %9 = urem i64 32000, %step.value
                                    %total.iterations.to.be.vectorized = sub i64 32000, %9
                                    %10 = insertelement <vscale x 4 x i64> poison, i64 %step.value, i64 0
                                   %stepVector.update.values = shufflevector <vscale x 4 x i64> %10, <vscale x
                                   ... 4 \times i64 poison, <vscale x 4 \times i32 zeroinitializer
                                    br label %vectorizing.block
                                 vectorizing.block:
                                 %11 = phi i64 [ 0, %Pre.Vectorization ], [ %25, %vectorizing.block ]
                                 %12 = phi <vscale x 4 x i64> [ %8, %Pre. Vectorization ], [ %26,
                                 ... %vectorizing.block ]
                                 %13 = getelementptr inbounds [32000 x i32], ptr @a, i64 0, i64 %11, !dbg !104
                                 %14 = getelementptr inbounds [32000 x i32], ptr @b, i64 0, i64 %11, !dbg !112
                                 %15 = load < vscale x 4 x i32 >, ptr %13, align 16
                                 %16 = load < vscale x 4 x i32 >, ptr %14, align 16
                                 %18 = getelementptr inbounds [32000 x i32], ptr @d, i64 0, i64 %11, !dbg !115
                                 %19 = getelementptr inbounds [32000 x i32], ptr @c, i64 0, i64 %11, !dbg !119
                                 %20 = call <vscale x 4 x i32> @llvm.aarch64.sve.ld1.nxv4i32(<vscale x 4 x
                                 ... i1> %17, ptr %18)
                                 %21 = call <vscale x 4 x i32> @llvm.aarch64.sve.mul.nxv4i32(<vscale x 4 x
                                ... i1> %17, <vscale x 4 x i32> %20, <vscale x 4 x i32> %16)
                                 \%22 = \text{call} < \text{vscale x 4 x i} 32 > \text{@llvm.aarch} 64.\text{sve.sub.nxv4i} 32 (< \text{vscale x 4 x})
                                ... i1> %17, <vscale x 4 x i32> %15, <vscale x 4 x i32> %21)
                                 %23 = call <vscale x 4 x i32> @llvm.aarch64.sve.ld1.nxv4i32(<vscale x 4 x
                                 ... i1> %17, ptr %19)
                                 %24 = call <vscale x 4 x i32> @llvm.aarch64.sve.add.nxv4i32(<vscale x 4 x
                                ... i1> %17, <vscale x 4 x i32> %22, <vscale x 4 x i32> %23)
                                 call void @llvm.aarch64.sve.st1.nxv4i32(<vscale x 4 x i32> %24, <vscale x 4
                                 ... x i1> %17, ptr %19)
                                 call void @llvm.aarch64.sve.st1.nxv4i32(<vscale x 4 x i32> %22, <vscale x 4
                                ... x i1 > \%17, ptr \%13)
                                 %25 = add i64 %step.value, %11
                                 %26 = add <vscale x 4 x i64> %12, %stepVector.update.values
                                 %terminate.condition = icmp uge i64 %25, %total.iterations.to.be.vectorized
                                 br i1 %terminate.condition, label %middle.block, label %vectorizing.block
                           middle.block:
                            %condition = icmp eq i64 %9, 0
                            br i1 %condition, label %for.cond.cleanup4, label
                           ... %Preheader.for.remaining.iterations
                                            Preheader.for.remaining.iterations:
                                            %27 = phi i64 [ 0, %for.cond2.preheader ], [ %25, %middle.block ]
                                            br label %for.body5
                                      for.body5:
                                       %indvars.iv = phi i64 [ %indvars.iv.next, %for.inc ], [ %27,
                                       ... %Preheader.for.remaining.iterations ]
                                       call void @llvm.dbg.value(metadata i64 %indvars.iv, metadata !82, metadata
                                       ...!DIExpression()), !dbg!91
                                       %arrayidx = getelementptr inbounds [32000 x i32], ptr @a, i64 0, i64
                                       ... %indvars.iv, !dbg !104
                                       %3 = load i32, ptr %arrayidx, align 4, !dbg !104, !tbaa !108
                                       %arrayidx7 = getelementptr inbounds [32000 x i32], ptr @b, i64 0, i64
                                       ... %indvars.iv, !dbg !112
                                       %4 = load i32, ptr %arrayidx7, align 4, !dbg !112, !tbaa !108
                                       %cmp8 = icmp sgt i32 %3, %4, !dbg !113
                                       br i1 %cmp8, label %if.then, label %for.inc, !dbg !114
                                                                                                F
                    %arrayidx14 = getelementptr inbounds [32000 x i32], ptr @d, i64 0, i64
                    ... %indvars.iv, !dbg !115
                    %5 = load i32, ptr %arrayidx14, align 4, !dbg !115, !tbaa !108
                    %mul = mul nsw i32 %5, %4, !dbg !117
                    %sub = sub nsw i32 %3, %mul, !dbg !118
                    call void @llvm.dbg.value(metadata i32 %sub, metadata !79, metadata
                    ... !DIExpression()), !dbg !86 %arrayidx16 = getelementptr inbounds [32000 x i32], ptr @c, i64 0, i64
                    ... %indvars.iv. !dbg !119
                    %6 = load i32, ptr %arrayidx16, align 4, !dbg !120, !tbaa !108
                    %add = add nsw i32 %sub, %6, !dbg !120
                    store i32 %add, ptr %arrayidx16, align 4, !dbg !120, !tbaa !108
                    store i32 %sub, ptr %arrayidx, align 4, !dbg !121, !tbaa !108
                    br label %for.inc, !dbg !122
                                         for.inc:
                                          %indvars.iv.next = add nuw nsw i64 %indvars.iv, 1, !dbg !123
                                          call void @llvm.dbg.value(metadata i64 %indvars.iv.next, metadata !82,
                                          ... metadata !DIExpression()), !dbg !91
                                          %exitcond.not = icmp eq i64 %indvars.iv.next, 32000, !dbg !124
                                         br i1 %exitcond.not, label %for.cond.cleanup4, label %for.body5, !dbg !92,
                                         ... !llvm.loop !125
                   for.cond.cleanup4:
                   %call19 = tail call i32 @dummy(ptr noundef nonnull @a, ptr noundef nonnull
                   ... @b, ptr noundef nonnull @c, ptr noundef nonnull @d, ptr noundef nonnull @e,
                  ... ptr noundef nonnull @aa, ptr noundef nonnull @bb, ptr noundef nonnull @cc,
                   ... i32 noundef 0) #8, !dbg !97
                   %inc21 = add nuw nsw i32 %nl.038, 1, !dbg !98
                   call void @llvm.dbg.value(metadata i32 %inc21, metadata !80, metadata
                   ...!DIExpression()),!dbg!89
                   %exitcond40.not = icmp eq i32 %inc21, 100000, !dbg !99
                   br i1 %exitcond40.not, label %for.cond.cleanup, label %for.cond2.preheader,
                   ...!dbg!90,!llvm.loop!100
                                                                               F
                                     Т
for.cond.cleanup:
%t2 = getelementptr inbounds %struct.args t, ptr %func args, i64 0, i32 1,
%call23 = tail call i32 @gettimeofday(ptr noundef nonnull %t2, ptr noundef
... null) #8, !dbg !94
```

entry:

%call24 = tail call i32 @calc checksum(ptr noundef nonnull @ func .s253) ... #8, !dbg !95 ret i32 %call24, !dbg !96

... !dbg !93